Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1-20. (Canceled)

21. (Currently Amended) A system, including:

a server, in communication with a first host and a second host, the first and second hosts executing a mobile application that jumps from the first host to the second host during execution and passes through the server,

the server storing, prior to a jump to the second host, a first instance of the mobile application, an instance of the mobile application including executable code for the mobile application,

the server receiving from the first host, during the jump to the second host, a second instance of the mobile application, and

the server detecting unwanted changes in contents of the mobile application including comparing the first and second instances.

22. (Previously Presented) The system of claim 21, wherein the contents are one or more from the group containing code, state data and itinerary data.

23. (Previously Presented) The system of claim 21, wherein the server detects unwanted changes responsive to receiving the mobile application from an untrusted host.

24. (Previously Presented) The system of claim 21, wherein the server stores the first instance of the mobile application responsive to the mobile application being received from a trusted host.

25. (Previously Presented) The system of claim 21, wherein the first instance includes a first checksum and the second instance includes a second checksum.

26. (Previously Presented) The system of claim 21, wherein the first instance includes a copy of the mobile application as it existed prior to the jump and the second instance includes a copy of the mobile application as it existed during the jump.

27. (Previously Presented) The system of claim 21, wherein the server forwards the mobile application to the second host.

28. (Currently Amended) A ~~centralized~~ method for verifying integrity of a jumping mobile application ~~at a location other than a dispatching host or a receiving host~~, the method including:

storing, prior to a jump and at a server, a first instance of a mobile application that jumps from a first host to a second host during execution, an instance of the mobile application including executable code for the mobile application;

receiving, during the jump and at the server, a second instance of the mobile application; and

detecting unwanted changes in contents of the mobile application, including the server comparing the first and second instances.

29. (Previously Presented) The method of claim 28, wherein the contents are one or more from the group containing code, state data and itinerary data.

30. (Previously Presented) The method of claim 28, wherein detecting unwanted changes includes detecting unwanted changes responsive to receiving the mobile application from an untrusted host.

31. (Previously Presented) The method of claim 28, wherein storing includes storing the first instance of the mobile application responsive to the mobile application being received from a trusted host.

32. (Previously Presented) The method of claim 28, wherein the first instance includes a first checksum and the second instance includes a second checksum.

33. (Previously Presented) The method of claim 28, wherein the first instance includes a copy of the mobile application as it existed prior to the jump and the second instance includes a copy of the mobile application as it existed during the jump.

34. (Previously Presented) The method of claim 28, further comprising:

forwarding the mobile application to the second host.

35. (Currently Amended) A computer program product including program instructions tangibly stored on a computer-readable medium and operable to cause a computer system to perform a method for verifying integrity of a jumping mobile application ~~at a location other than a dispatching host or a receiving host~~, the method including:

storing, prior to a jump and at a location other than a first host or a second host, a first instance of a mobile application that jumps from the ~~a~~ first host to the ~~a~~ second host during execution, an instance of the mobile application including executable code for the mobile application,

receiving, during the jump and at the location, a second instance of the mobile application, and

detecting unwanted changes in contents of the mobile application including comparing, at the location, the first and second instances.

36. (Previously Presented) The computer program product of claim 35, wherein the contents are one or more from the group containing code, state data and itinerary data.

37. (Previously Presented) The computer program product of claim 35, wherein detecting unwanted changes includes detecting unwanted changes responsive to receiving the mobile application from an untrusted host.

38. (Previously Presented) The computer program product of claim 35, wherein storing includes storing the first instance of the mobile application responsive to the mobile application being received from a trusted host.

39. (Previously Presented) The computer program product of claim 35, wherein the first instance includes a first checksum and the second instance includes a second checksum.

40. (Previously Presented) The computer program product of claim 35, wherein the first instance includes a copy of the mobile application as it existed prior to the jump and the second instance includes a copy of the mobile application as it existed during the jump.

41. (Previously Presented) The computer program product of claim 35, further comprising:

forwarding the mobile application to the second host.

42. (New) The system of claim 21, wherein the server is operable to:

in response to receiving a request from the mobile application for code for execution on the second host, determine whether the first host is allowed to inject code;

when the first host is determined as being allowed to inject code, retrieve the code from the first host and send the code to the mobile application; and

when the first host is determined as not being allowed to inject code, search for the code on the server and, if the code is found on the server, send the code found on the server to the mobile application.

43. (New) The system of claim 21, wherein the server is operable to:

determine whether the first host is allowed to inject code and whether the mobile application has been previously dispatched;

when the first host is determined as not being allowed to inject code and the mobile application is determined to have not been previously dispatched, remove the mobile application's code;

when the first host is determined as not being allowed to inject code, the mobile application is determined to have been previously dispatched, restore the mobile application's datastore as the datastore existed for the previous dispatch; and

when the first host is determined as being allowed to inject code, determine whether a host originating the mobile application is trusted and, when the host originating the mobile application is trusted, dispatch the mobile application.